Proceedings of the American Control Conference
Philadelphia, Pennsylvania • June 1998

Nonlinear Model Predictive Control of the Tennessee Eastman Process

Alex Zheng, zzheng@ecs.umass.edu
Department of Chemical Engineering
University of Massachusetts
Amherst, MA 01003-3110

Abstract

The main purposes of this paper are to illustrate several key issues in the implementation of a conventional Nonlinear Model Predictive Control algorithm on a reasonably large industrial process and to test the effectiveness of the Nonlinear Model Predictive Control algorithm recently proposed by Zheng for control of large nonlinear systems with constraints. We show why a conventional Nonlinear Model Predictive Control algorithm may fail to provide integral control under very reasonable conditions (i.e., integral control is guaranteed if and only if a global solution is implemented and the output horizon is infinite) and illustrate this undesirable behavior through simulations on the Tennessee Eastman process. In addition to computational advantage, we argue that Zheng's algorithm may be preferred based on robust performance consideration.

1. Introduction

While many Nonlinear Model Predictive Control (NMPC) algorithms have been proposed for control of nonlinear systems with constraints, their practical implementation faces many obstacles. In general, an NMPC algorithm solves an on-line optimization problem at each sampling time. In contrast to linear MPC where usually a quadratic program needs to be solved, the on-line optimization problem is generally non-convex for NMPC. Thus, practical implementation of NMPC becomes difficult for any reasonably nontrivial nonlinear systems.

To overcome this difficulty, Zheng [14] recently proposed a NMPC algorithm which combines the best features of an exact optimization approach [1, 7, 4, 8, 2, etc.] and approximation approach [5, 6, etc.]. The basic idea is to compute only the first control move, which is actually implemented, exactly, while approximating the rest of control moves, which are never implemented. The purpose of this paper is to test this algorithm on a realistic and reasonably large industrial process and compare its performance to a conventional NMPC algorithm which attempts to solve the on-line optimization problem exactly.

The paper is organized as follows: Section 2 describes the basic ideas behind a conventional NMPC algorithm and the NMPC algorithm proposed by Zheng. The Tennessee Eastman (TE) process is briefly described in Section 3. Simulation studies are described in Section 4. Section 5 concludes the paper.

2. NMPC Algorithms

We consider a continuous-time nonlinear system described below.

$$\begin{cases} \dot{x}(t) &= f(x(t), u(t), d(t)) \\ y(t) &= g(x(t), d(t)) \end{cases} \tag{1}$$

where x of dimension n_x is the state, u of dimension n_u is the input, y of dimension n_y is the output, and d of dimension n_d is the disturbance.

The objective function at sampling time k is defined as follows:

$$\Phi_k = \int_0^{H_p} ||r(k+t) - y(k+t|k)||^2_{\Gamma_y} dt + \sum_{i=0}^{H_c-1} ||\Delta u(k+i|k)||^2_{\Gamma_u} \tag{2}$$

where H_p is the output horizon, H_c is the input (or control) horizon, k is the k^{th} sampling time (the sampling period is assumed to be one for simplicity), r is the desired output trajectory, $\Delta u(k) \triangleq u(k) - u(k-1)$, and $\Gamma_y > 0$ [1] and $\Gamma_u > 0$ are diagonal weights. $||x||_P = \sqrt{x^T P x}$ denotes the weighted 2-norm. Notice that the inputs are discrete (i.e., $u(t)$

[1] In general, the output weight Γ_y depends on the predicted output $y(k+t|k)$. For example, the following definition allows

is constant for $t \in [k, k+1))$ while the outputs are continuous. Δu is penalized instead of u to generate integral control, at least when the system is *linear* (see discussion in Section 4). A conventional NMPC algorithm is defined as follows.

Definition 1 (Controller NMPC #1) *At each sampling time k, $\Delta u(k)$ equals the first control move $\Delta u(k|k)$ of the sequence $\{\Delta u(k|k), \cdots, \Delta u(k+H_c-1|k)\}$ which minimizes the following.*

$$\min_{\Delta u(k|k),\cdots,\Delta u(k+H_c-1|k)} \quad \Phi_k \qquad \text{subject to}$$

$$\left\{\begin{array}{l} \dot{\hat{x}}(k+t|k) = f(\hat{x}(k+t|k), u(k+t|k), \hat{d}(k+t|k)) \\ y(k+t|k) = g(\hat{x}(k+t|k), \hat{d}(k+t|k)), t \geq 0 \\ u(k+i|k) \in \mathcal{U}, i = 0, \cdots, H_c - 1 \\ \Delta u(k+i|k) = 0, i \geq H_c \end{array}\right. \tag{4}$$

where $\bullet(k+i|k)$ denotes the variable at time $k+i$ predicted at time k, $\hat{\bullet}(k)$ denotes the estimated variable at time k, and the objective function Φ_k is defined by (2).

Under some reasonable assumptions, it is straightforward to prove that asymptotic stability of the closed loop system can be guaranteed if and only if the optimization problem is feasible. The optimization problem (4) has $n_u H_c$ decision variables. In general, for reasons of stability and performance, H_c should be chosen reasonably large [10, 15]. Since the optimization problem (4) is generally non-convex, the on-line computational demand grows exponentially with the number of decision variables. Thus, practical implementation of `Controller NMPC #1` becomes difficult, if not impossible, for any reasonably large nonlinear system.

Motivating by this difficulty, Zheng [14] recently proposed a NMPC algorithm which combines the best features of NMPC algorithms which attempt to solve the problem exactly and NMPC algorithms which reduce the optimization problem to a quadratic program via linearization. The basic idea is to compute the first control move $\Delta u(k|k)$, which is implemented, exactly, while approximating the rest of control moves, which are never implemented, via linearization. Furthermore, in contrast to almost all the NMPC algorithms which use an open-loop control strategy, a closed loop control strategy is incorporated into the algorithm. The algorithm is stated below.

Definition 2 (Controller NMPC #2) *At each sampling time k, $\Delta u(k)$ is the solution to the following optimization problem.*

$$\min_{\Delta u(k|k)} \quad \Phi_k \qquad \text{subject to}$$

$$\left\{\begin{array}{l} \dot{\hat{x}}(k+t|k) = f(\hat{x}(k+t|k), u(k+t|k), \hat{d}(k+t|k)) \\ y(k+t|k) = g(\hat{x}(k+t|k), \hat{d}(k+t|k)) \\ u(k|k) \in \mathcal{U} \\ \mathbf{\Delta u(k+i|k)} = \mathrm{sat}(\mathbf{C(x(k+i|k), e(k+i|k))}) \\ \qquad \mathbf{i = 1, \cdots, H_p - 1} \end{array}\right. \tag{5}$$

where sat *is the saturation function and $e(k+i|k) \triangleq r(k+i) - y(k+i|k)$.*

Notice that *all* the future control moves (i.e., $\Delta u(k+i|k), i \geq 1$) are approximated by the controller C which can be computed analytically for a linear MPC without constraints. The on-line optimization problem for `Controller NMPC #2` has *only* n_u decision variables, *regardless* of the control horizon H_c, resulting in significant savings in on-line computational time (Figure 1). Under reasonable assumptions, asymptotic stability of the closed loop system with `Controller NMPC #2` is guaranteed if and only if the optimization problem is feasible [14]. The rest of this paper focuses on comparing the two algorithms on the TE process.

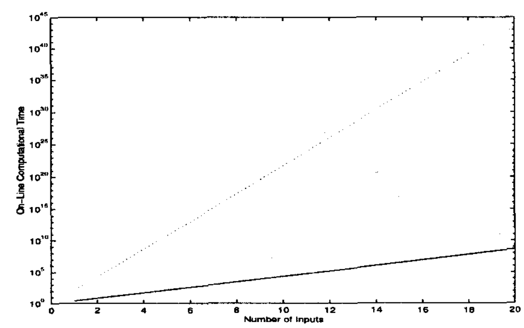

Figure 1: Comparison of on-line computational demand for `Controller NMPC #1` and `#2` with $H_c = 5$ (solid: `Controller NMPC #2`; dotted: `Controller NMPC #1`).

---

us to handle the output constraints as "soft" constraints.

$$\Gamma_y(y)_{ii} = \left\{\begin{array}{ll} \Gamma_y(0)_{ii}[1+\epsilon_i(y_i - y_i^{\min})^2] & \text{if } y_i \leq y_i^{\min} \\ \Gamma_y(0)_{ii} & \text{if } y_i^{\min} \leq y_i \leq y_i^{\max} \\ \Gamma_y(0)_{ii}[1+\epsilon_i(y_i - y_i^{\max})^2] & \text{if } y_i \geq y_i^{\max} \end{array}\right. \tag{3}$$

Details are discussed in [14].

Manipulated Variables	Controlled Variables
Feed 1 (pure A)	*Reactor pressure*
Feed 2 (pure D)	Reactor liquid level
Feed 3 (pure E)	Separator liquid level
Feed 4 (A & C)	Stripper reboiler level
Recycle flow	*H in product*
Purge	*Production rate*
Separator underflow	
Stripper underflow	
Reactor temperature	
Separator temperature	

Table 1: Summary of manipulated and controlled variables.

3. The TE Process

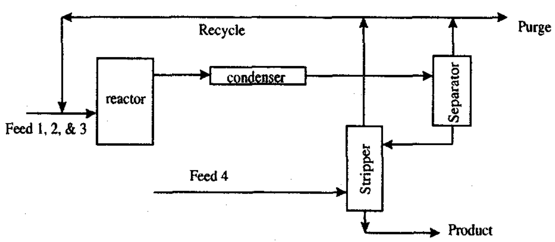

Figure 2: Simplified block diagram of the TE problem.

The TE process was proposed by Downs and Vogel [3] for testing alternative control and optimization strategies for continuous chemical processes (Figure 2). A simplified model, which has 26 states and consists of two PI controllers,[2] developed by Ricker [11] is used in this study. Ten inputs are selected as manipulated variables and six outputs as controlled variables. They are listed in Table 1. Three level controllers, with parameter values from [11], are implemented on the model so that the resulting system is stable. Thus, NMPC algorithms handle a subsystem with seven inputs and three outputs, which are italicized in Table 1. Space limitation prevents us from any further discussion on the TE process; but the interested readers are referred to the papers by Downs and Vogel [3] and Ricker [11] for details.

[2]The two PI controllers control reactor and separator temperatures by manipulating reactor and condenser coolant valves, respectively.

4. Simulation Studies

A sampling time of 0.25 hours and $H_p = 800$ (i.e., 200 hours) are used in all the simulations, except noted otherwise. The input and output weights are chosen by scaling all the variables so that their acceptable values are within ± 1.

$$\begin{aligned}\Gamma_u &= \lambda\,\mathrm{diag}([90\ \ 15\ \ 12\ \ 4\ \ 1\ \ 66\ \ 100])\\ \Gamma_y &= \mathrm{diag}([1\ \ 400\ \ 20])\end{aligned}$$

Here λ is an adjustable parameter.

While many possible changes can be made, we decide to carry out all the comparisons on a setpoint change from 43.8%H in product to 53%. The reasons are (a). we felt that with such servo change the comparisons are more meaningful since important issues such as disturbance filtering can be ignored and (b). process behavior corresponding to these two operating modes are quite different (i.e., nonlinearity becomes important).

The on-line optimization problem is non-convex as the solution was found to depend on the initial guess. To have a meaningful comparison, a global solution needs to be determined at each sampling time. However, this is *not* feasible for `Controller NMPC #1` for any $H_c > 1$ as it takes *several days* to solve the optimization problem at each sampling time for $H_c = 2$. On the other hand, any comparison of the two algorithms requires $H_c > 1$ since `Controller NMPC #1` reduces to `Controller NMPC #2` for $H_c = 1$ in the special case where $C = 0$. To overcome this difficulty, a "global" solution is determined as the best solution among optimal solutions for different initial guesses. While it is possible to determine a global solution for `Controller NMPC #2`, we have used the same method to determine a "global" solution for `Controller NMPC #2`. The Powell algorithm is used to determine a local optimal solution.

While C can be any controller (or controllers) that approximates the future control moves, for simplicity, C is chosen to be a linear controller corresponding to `Controller NMPC #1` with a linearized model of the nonlinear system at the nominal operating conditions. Specifically, C equals the controller **nlmpcsim** in MPC Toolbox [9]. However, it is approximated by a continuous controller to speed up the simulation. The Gear's method in SIMULINK is used for integration. It should be emphasized that both the optimization (i.e., Powell) and integration (i.e., Gear) methods are merely chosen for convenience and that they may not be optimal choices.

4.1. Integral Control

The importance of integral control in process control is well recognized. For linear systems, `Controller NMPC #1` reduces to linear MPC which guarantees integral control for *almost* all values of tuning parameters. However, this is *not* the case for nonlinear systems. It is straightforward to show that `Controller NMPC #1` *guarantees* integral control if and only if a global solution is found *and* $H_p = \infty$,[3] both of which are impractical.

The `if` part of this statement is obvious: Feasibility of zero offset implies that the value function at the first sampling time is finite. A Lyapunov argument can then be used to prove that the output approaches the setpoint asymptotically. The `only if` part can be proven as follows: Clearly an offset may exist if only a local solution is found. If H_p is finite, then the value function is finite for an infinite number of input values (which is only true for several steady-state input values if $H_p = \infty$). For some nonlinear function, a global solution does not necessarily correspond to its steady-state value. To illustrate this point, let us consider a simple system, $y(t) = u(t)^3$. Let the objective function be $(r-y)^2 + \lambda \Delta u^2$. Suppose $y = 0$ initially. It is simple to show that the global solution is $\Delta u = 0$ if $\lambda > (\frac{27r^4}{16})^{\frac{1}{3}}$, from which we conclude that there is always an offset if $\lambda > 0$!

Both of these cases are observed in the simulations. Figures 3 and 4 show the existence of an offset due to local optima and finite output horizon, respectively. It should be noted that the results in Figure 4 are based on 1,000 different initial guesses. One way to resolve the difficulty of guaranteeing integral control is to design a linear controller with integral control underneath `Controller NMPC #1`, which is precisely what `Controller NMPC #2` does. Thus, for this reason (in addition to computational advantage), `Controller NMPC #2` may be preferred.

4.2. Performance Comparison

There are two differences between `Controller NMPC #1` and `Controller NMPC #2`: (a). Future control moves in `Controller NMPC #2` are approximated by the controller C but are computed exactly in `Controller NMPC #1`; and (b). `Controller NMPC #1` assumes $\Delta u(k+i|k) = 0, i \geq H_c$, while `Controller NMPC #2` calculates these control moves via the controller C. These differences imply that `Controller NMPC #1` should always outperform `Controller NMPC #2` for sufficiently large values of H_c (i.e., the first difference dominates). However, we expect the difference in performance becomes small when the input weight becomes large. The reason is simply that the control action is small and that the system stays near where it is linearized (i.e., linear approximation is adequate). Unfortunately, the question of how much better `Controller NMPC #1` performs than `Controller NMPC #2` can not be answered due to excessive simulation time with `Controller NMPC #1` for $H_c > 2$. The second difference implies that `Controller NMPC #2` *may* outperform `Controller NMPC #1` for small values of H_c, especially for large input weights or when linear approximation is adequate.

The above observations are confirmed by simulations. Figure 5 compares performance for the two controllers for two values of λ, 0 and 2000. For $\lambda = 0$, `Controller NMPC #1` outperforms `Controller NMPC #2`. For $\lambda = 2000$, `Controller NMPC #2` outperforms `Controller NMPC #1` slightly. As expected, the performance for $\lambda = 2000$ is more sluggish than that for $\lambda = 0$.

Because of model uncertainty, which always ex-

[3] We have assumed that zero offset is possible.

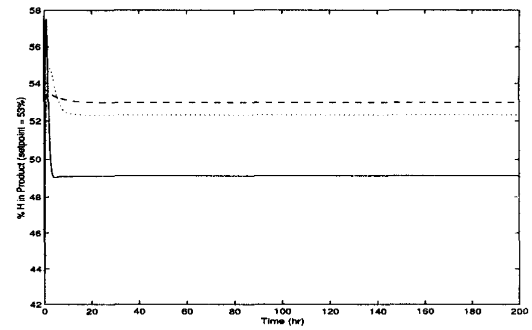

Figure 3: Offset due to local optima.

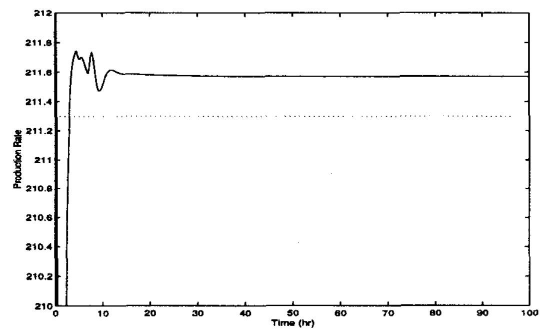

Figure 4: Offset due to finite output horizon ($H_p = 200, H_c = 1, \lambda = 1000$).

ists, the input weight should be selected reasonably large to ensure robustness. Therefore, one may argue that `Controller NMPC #2` may be preferred based on robust performance consideration. Another reason that `Controller NMPC #2` may be preferred in the robust case is that a closed loop control strategy is incorporated into the algorithm through the controller C. We are currently in the process of testing this observation for the TE process. Specifically, we are using the model for the TE process originally developed by Down and Vogel as the actual plant. We are also planning to use the estimator developed by Ricker and Lee [12] for estimation.

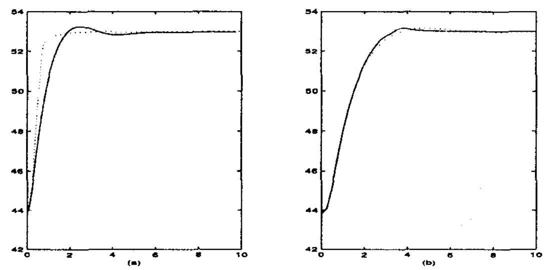

Figure 5: Effects of λ on approximation (solid: `Controller NMPC #2`; dotted: `Controller NMPC #1`). (a). $\lambda = 0$; (b). $\lambda = 2000$.

5. Conclusions

We have tested the NMPC algorithm proposed by Zheng on the TE process and compared its performance to a conventional NMPC algorithm which attempts to solve the on-line optimization exactly. We argue and illustrate through simulations that Zheng's algorithm may be preferred based on considerations of on-line computational demand, integral control, and robust performance.

References

[1] L. T. Biegler and J. B. Rawlings. Optimization approaches to nonlinear model predictive control. In *Conf. Chemical Process Control (CPC-IV)*, pages 543–571, South Padre Island, Texas, 1991. CACHE-AIChE.

[2] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 1997. in press.

[3] J. J. Downs and E. F. Vogel. A plant-wide industrial-process control problem. *Computers and Chemical Engineering*, 17(3):245–255, 1993.

[4] J. W. Eaton and J. B. Rawlings. Model-predictive control of chemical processes. *Chem. Eng. Sci.*, 47:705–720, 1992.

[5] C. E. García. Quadratic dynamic matrix control of nonlinear processes. An application to a batch reactor process. In *AIChE Annual Meeting*, San Francisco, CA, 1984.

[6] J. H. Lee and N. L. Ricker. Extended Kalman filter based on nonlinear model predictive control. *Ind. Eng. Chem. Res.*, 33:1530–1541, 1994.

[7] D. Q. Mayne. Nonlinear model predictive control: An assessment. In *CPC V*, Lake Tahoe, CA, 1996.

[8] H. Michalska and D. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Aut. Control*, 38(11):1623–1633, November 1993.

[9] M. Morari and N. L. Ricker. *Model Predictive Control Tooolbox: For Use with MATLAB*. The MathWorks Inc., Natick, MA, May 1995.

[10] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Trans. Aut. Control*, 38(10):1512–1516, October 1993.

[11] N. L. Ricker. Decentralized control of the tennessee eastman challenge process. *Journal of Process Control*, 6:205–221, 1996.

[12] N. L. Ricker and J. H. Lee. Nonlinear model predictive control of the Tennesee Eastman challenge process. *Comp. and Chem. Eng.*, 19(9):961–981, 1995.

[13] A. Zheng. A computationally efficient nonlinear model predictive control algorithm. In *Proceedings of American Control Conf.*, Albuquerque, NM, 1997.

[14] A. Zheng. A computationally efficient nonlinear model predictive control algorithm with guaranteed stability. In *NATO-ASI*, Antalya, Turkey, 1997.

[15] A. Zheng, V. Balakrishnan, and M. Morari. Constrained stabilization of discrete-time systems. *International Journal of Robust and Nonlinear Control*, 5(5):461–485, Aug. 1995.